시멘틱 웹상의 RDF 데이터 관리 시스템[1]

서명희° 안재용 민준기 정진완
KT 서비스개발 연구소°, 한국과학기술원 전자전산학과
mhseo@kt.co.kr°, {jyahn, jkmin, chungcw}@kaist.ac.kr

An RDF Data Management System On The Semantic Web

Myounghee Seo° Jaeyoung Ahn Junki Min Chin-wan Chung
KT Service Development Institute°,
Division of Computer Science, Dept. of Computer Science & Electrical Engineering, KAIST

요 약

시멘틱 웹상에서는 정보 리소스들이 서로 의미적으로 연결되어, 이를 컴퓨터가 처리할 수 있다. Resource Description Framework (RDF)는 이런 의미적 연결성을 제공한다. 시멘틱 웹이 발전하기 위해서는 RDF 데이터를 효율적으로 관리하기 위한 방법이 매우 중요하다. 본 논문에서는 RDF 데이터를 XML 데이터베이스 시스템에 저장하고 이를 검색하는 기법을 제안한다. XML 데이터베이스 시스템을 사용함으로써 XML 데이터와 RDF 데이터를 통합적이고 효율적으로 관리할 수 있다. 또한, 효율적인 검색 방법과 성능을 향상시킬 수 있는 방법들을 제안하고 있다. 논문에서 제안한 질의 처리 기법은 기존 연구보다 나은 성능을 보여준다.

1 서 론

최근 차세대 웹으로 시멘틱 웹(Semantic Web)이 부각되고 있다. 기존의 월드 와이드 웹과는 달리 시멘틱 웹상에서는 정보 리소스들의 의미가 정의되어 있고, 이들간의 의미적 연결을 지원한다. 시멘틱 웹에서 이런 의미적 연결성을 지원하기 위해 Resource Description Framework(RDF)[1]를 사용한다. RDF는 웹 리소스들의 메타 데이터를 표현하기 위한 데이터 모델이다. RDF를 이용해서 정의된 데이터를 시멘틱 웹상에서 자유롭게 접근할 수 있고, 이런 정보를 컴퓨터가 처리하여 원하는 정보를 얻을 수 있다.

시멘틱 웹이 차세대 웹으로 자리 잡기 위해서는 RDF 데이터를 잘 다루기 위한 기술들이 선행 연구되어야 한다. 특히, 방대한 양의 RDF 데이터를 효율적으로 저장하고 검색하는 기법이 중요하다고 할 수 있다.

본 연구에서는 RDF 데이터를 XML 데이터베이스 시스템에 저장하고, RDF 질의 언어인 RQL[3]을 이용하여 검색하는 방법을 제안하고 있다. 기존의 연구와 달리, RDF 데이터를 XML 데이터베이스 시스템에 저장함으로써 XML 데이터와 이의 메타 데이터인 RDF 데이터를 같이 저장할 수 있어 이들을 통합적이고 효율적으로 관리할 수 있다. 또한 RDF 데이터를 XML 형태로 표현할 수 있어, XML 데이터베이스 시스템에 RDF 데이터를 별도의 처리과정 없이 XML 형태 그대로 저장할 수 있다.

또한, 본 연구에서는 RDF 데이터에 대한 RQL 질의를 XML 질의 언어인 XPath[5] 질의로 변환하여 처리하는 효율적인 방법을 제안하고 있다. 또한, 질의 처리 과정에서 검색 성능을 향상시킬 수 있는 몇 가지 방법들을 사용하였다.

2 관련 연구

이제까지 연구된 RDF 저장 시스템으로는 ICS-FORTH의 RDFSuite[2] 등이 있다. RDFSuite 시스템은 RDF 데이터를 객체 관계형 데이터베이스 시스템에 저장하고 RQL 질의가 들어오면 이를 SQL3로 변환하여 결과를 가져온다. 이 시스템은 기반 데이터베이스 시스템으로 객체 관계형 데이터베이스를 사용하였으므로, 저장할 때 데이터베이스 스키마에 맞게 데이터를 구성하여 저장하여야 하고, 검색할 때에도 여러 테이블 간의 조인 등의 오버헤드가 있다.

본 논문에서 제안하는 시스템은 기반 데이터베이스 시스템으로 XML 데이터베이스 시스템을 사용하여, 저장할 때, 별도의 처리 없이 데이터 그대로 저장할 수 있다. 또한, XML 데이터와 이의 메타 데이터인 RDF 데이터를 같은 시스템에 저장하고 관리함으로써 두 데이터 간의 공간적 차이를 줄여, 통합적이고 효율적인 관리가 가능하다.

3 RDF/RDF Schema 데이터 저장

3.1 RDF 데이터 저장 과정

RDF 데이터 저장 과정은 그림 1과 같다. RDF description 데이터와 RDF schema[4] 데이터가 들어오면 이를 Validator에서 구문과 의미에 대한 검증을 수행한다. 그 다음으로, Standardizer에서 여러 형태의 RDF/XML 구문[6]을 저장 공간의 효율성과 검색의 용이성을 위해 하나의 구문으로 통합하는 작업을 수행한다. 통일된 형태의 standard RDF/XML 데이터가 XML 데이터베이스 시스템에 저장된다. 이때, class/property hierarchy extractor에서 RDF schema 데이터의 클래스와 속성의 계층 구조 및 속성의 domain, range 제약에 대한 정보를 뽑아서 별도의 XML 문서로 저장한다. 이렇게 별도로 저장된 정보들은 검색의 성능을 향상시키기 위해 사용된다.

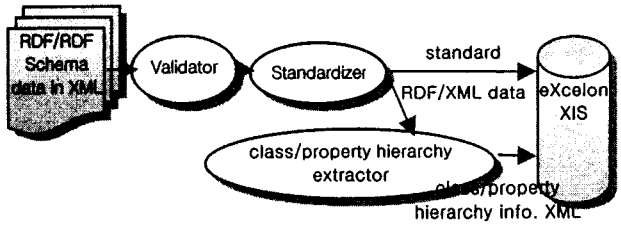

그림 1. RDF 데이터 저장 과정

3.2 RDF/XML 형태 통일 (Standard RDF/XML Data)

RDF description 데이터 및 RDF schema 데이터는 RDF/XML 구문[6]을 이용하여 여러 형태의 XML 데이터로 표현될 수 있다. 또한, XML 질의 언어인 XPath 질의는 XML 구문에 의존적이기 때문에, 입력된 RDF/XML 데이터를 통일된 형태의 RDF/XML 데이터로 변환할 필요가 있다. 본 연구에서는 검색의 용이성과 저장의 효율성을 위해 RDF/XML 데이터의 여러 단축(abbreviation) 구문[6] 중에서 두 가지 구문만을 적용한 형태로 모든 RDF 데이터를 통일하였다.

3.3 Class/Property Hierarchy Extractor

RQL은 RDF schema 클래스 및 속성의 트랜지티브 클로저(transitive

[1] 본 연구는 정보통신부의 대학 IT 연구센터(ITRC) 지원을 받아 수행되었습니다.

closure)를 지원한다. 또한, 속성의 domain, range 제약에 대한 질의도 지원한다. 이런 질의들을 RDF/XML 형태의 RDF schema 데이터에서 검색하는 작업은 간단하지가 않다. 특히, 클래스 또는 속성의 계층 깊이가 깊어질수록 계층 정보를 얻기 위해서 더 많은 XPath 질의가 필요하게 된다. 따라서, 본 연구에서는 클래스 및 속성의 계층 구조와 속성의 domain, range 정보를 축출하여 XML 형태로 저장하여, 이 정보에 대한 질의를 간단한 XPath 질의로 수행될 수 있도록 하였다.

4 RDF/RDF Schema 데이터 검색

4.1 질의 처리 과정

RQL 질의를 처리하는 과정은 그림 2와 같다. 사용자로부터 RQL 질의가 들어오면, 이를 Parser에서 파싱한 후 질의의 분류에 따라 해당 질의 처리기가 RQL 질의를 XPath 질의로 변환하여 XML 데이터베이스 시스템인 eXcelon XIS 시스템에 보내어 결과를 가져온 후, 결과 생성기가 결과를 테이블 형태로 변환하여 사용자에게 보내주게 된다. 질의 처리 시 RDF 데이터를 저장할 때 뽑아둔 RDF schema 클래스와 속성의 계층 정보를 사용하여 질의를 효율적으로 처리한다.

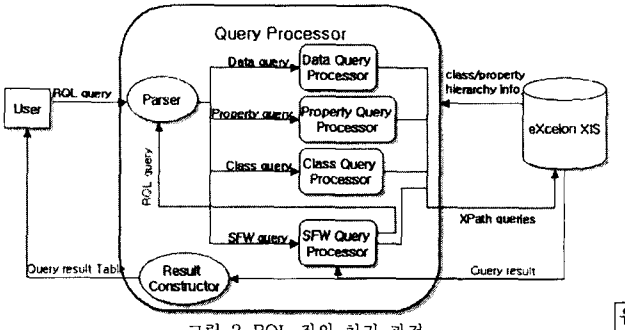

그림 2. RQL 질의 처리 과정

RQL 질의는 형태상 크게 두 가지 - Select-From-Where(SFW) Query, Non SFW Query - 로 분류할 수 있다. Non SFW Query는 다시 몇 개의 하위 질의로 분류할 수 있다.

4.2 Non Select-From-Where(Non SFW) Query 변환

표 1.는 Non-SFW query의 분류에 대해 각각에 해당되는 RQL 질의의 예와 각각을 XPath 질의로 변환한 예를 보여주고 있다. 표에서 각각의 질의에 대해 위 부분은 질의의 예이고, 아래 부분은 이를 XPath 질의로 변환한 형태를 보여 준다. 변환된 질의에서 rdf_data.xml은 RDF 데이터가 저장되어 있는 XML 문서의 이름이고, classHierarchy.xml은 클래스 계층 구조 정보가 저장되어 있는 XML 문서의 이름이다. Non-SFW query는 미리 축출된 정보를 이용하여 간단한 XPath질의로 변환할 수 있다.

표 1. Non-SFW Query의 XPath로의 변환

Class Query	TypeOf	typeOf(*www.culture.net#picasso132*)
		document("*rdf_data.xml*")//rdf:Description[@rdf:about="*www.culture.net#picasso132*"]/rdf:type/@rdf:resource
	SuperClassOf	SuperClassOf(*http://www.icom.com/schema1.rdf#Painter*)
		document("*classHierarchy.xml*")//*NS:Painter*/ancestor::*
	Domain	domain(*http://www.icom.com/schema1.rdf#creates*)
		document("*propertyHierarchy.xml*")//*NS:creates*/@rdf:domain
	URI	http://www.w3.org/2000/01/rdf-schema#Class
		document("*classHierarchy.xml*")//*
Proper	SuperPropertyOf	SuperPropertyOf(*http://www.icom.com/schema1.rdf#sculpts*)
		document("*propertyHierarchy.xml*")//*NS:sculpts*/ancestor::*
	SubPropertyOf	SubPropertyOf(*http://www.icom.com/schema1.rdf#creates*)
	Of	document("*propertyHierarchy.xml*")//*NS:creates*/descendant::*
	URI	http://www.w3.org/1999/02/22-rdf-syntax-ns#Property
		document("*propertyHierarchy.xml*")//*
C	InstanceOf	^*http://www.icom.com/schema1.rdf#Painter*
		document("*rdf_data.xml*")//rdf:Description[rdf:type/@rdf:resource="*http://www.icom.com/schema1.rdf#Painter*"]/@rdf:about
	URI	*http://www.icom.com/schema1.rdf#Artist*
		∃*c*∈SubClassOf(*http://www.icom.com/schema1.rdf#Artist*), document("*rdf_data.xml*")//rdf:Description[rdf:type/@rdf:resource=*c*]/@rdf:about

4.3 Select-From-Where (SFW) Query 처리

SFW query 처리 과정을 간단하게 살펴보면, 우선 SFW query를 select, from, where부분으로 나눈다. 그 후에 from절의 경로 표현식을 단순 경로식들로 분리한다. 그 다음으로 where절에 있는 조건(condition)을 and로 연결된 형태(conjunctive form)로 바꾼 후, 이를 and를 기준으로 분리한다. 그 다음 작업으로는 from절의 각각의 경로식을 XPath 질의로 변경한다. 이때, where절의 조건 중에서 같이 처리할 수 있는 것들은 XPath 질의의 프리디킷(predicate)으로 넣어, XPath 질의에 대한 결과를 가져올 때, 조건에 맞는 데이터만을 가져오게 한다. 조건 중에서 같이 처리할 수 없는 것들은 해당 경로식의 결과를 가져온 후, 조건에 맞는 데이터만을 가려낸다. 마지막 작업으로 각각의 경로식에 대한 결과값들을 조인한다. 표 2는 SFW query의 각각의 경로 표현식에 대해 XPath 질의로 변환과정을 보여준다.

표 2. RQL SFW Query의 경로 표현식의 분류 및 해당 XPath 질의

유형	경로 표현식	변환된 XPath 질의
Data path	*c*{X}	∃*c'*∈SubClassOf(*c*), <//rdf:Description[type/@resource=*c'*/@about>
	$X{Y}	∃*c*∈C, <*c*,//rdf:Description[rdf:type/@rdf:resource=*c*]/@rdf:about>
	{X}*p*{Y}	∃*p'*∈SubPropertyOf(*p*), V1={//rdf:Description[*p*]/@rdf:about}, ∃*v1*∈V1,<*v1*,//rdf:Description[@rdf:about=*v1*]/*p*/@rdf:resource>
	{X}@P{Y}	∃*p*∈P, V1={//rdf:Description[*p*]/@rdf:about}, ∃*v1*∈V1,<*v1*,*p*,//rdf:Description[@rdf:about=*v1*]/*p*/@rdf:resource>
Schema path	*Class*{X}	∃*c*∈C, <*c*>
	Property{P}	∃*p*∈P, <*p*>
	c{:$C}	∃*c'*∈SubClassOf(*c*), <*c'*>
	$X{:$Y}	∃*c*∈C, <*c*, SubClassOf(*c*)>
	{:$X}*p*{:$Y}	∃*c1*∈SubClassOf(Domain(*p*)), ∃*c2*∈SubClassOf(Range(*p*)), <*c1*, *c2*>
	{:$X}@P{:$Y}	∃*p*∈P, ∃*c1*∈SubClassOf(Domain(*p*)), ∃*c2*∈SubClassOf(Range(*p*)), <*c1*,*p*,*c2*>
Mixed path	*c*{X:$C}	∃*c'*∈SubClassOf(*c*), <//rdf:Description[rdf:type/@rdf:resource=*c'*]/@rdf:about, *c'*>
	{X:$Z}*p*{Y:$W}	∃*p'*∈SubPropertyOf(*p*), V1={//rdf:Description[*p*]/@rdf:about}, ∃*v1*∈V1, C1={//rdf:Description[@rdf:about=*v1*]/rdf:type/@rdf:resource}, V2={//rdf:Description[@rdf:about=*v1*]/*p'*/@rdf:resource}, ∃*v2*∈V2, C2={//rdf:Description[@rdf:about=*v2*]/rdf:type/@rdf:resource}, ∃*c1*∈C1, *c1*∈SubClassOf(Domain(*p*)), ∃*c2*∈C2, *c2*∈SubClassOf(Range(*p*)), <*v1*, *c1*, *v2*, *c2*>

4.3.1 Where절의 조건(condition) 처리 방법

표 4는 조건의 비교 연산과 not연산에 대해 XPath 프리디킷으로 변환가능한지 그리고 가능하다면 어떻게 변환할 수 있는지 보여주고 있다. 이렇게 가능한 조건을 XPath 질의의 프리디킷으로 표현하여 조건

에 대한 연산을 가능한 데이터베이스 시스템에서 처리하도록 함으로써, 데이터베이스 시스템의 질의 처리기를 최대한 활용할 수 있고, 데이터베이스에서 주고 받는 데이터의 양을 줄일 수 있어 RQL 질의 처리를 효율적으로 수행하도록 하였다.

표 3. 조건 연산의 분류에 따른 XPath 질의의 프리디킷으로의 변환

	<, <=, >, >=	=, !=, not	like
		=, !=, not	contains()

4.3.2 조인(join) 연산 처리

각각의 경로 표현식의 결과를 순서대로 둘씩 조인을 수행한다. 두개의 결과가 조인이 필요 없는 경우는 두 결과를 프로덕트하고, 조인이 필요한 경우는 공통의 변수에 대해 sort-merge 조인을 수행한다. 이런 조인 연산을 통해서 데이터베이스 시스템에 대한 접근을 최소화하였고, 각각의 결과에 대한 조인 횟수를 줄여 조인 연산의 오버헤드를 줄였다.

5 실험

실험에서 4개의 질의를 사용하였다. 첫번째 질의(Q1)는 RDF schema 데이터를 검색하는 schema query 중에서 하위 클래스들을 검색하는 질의이고, 두 번째 질의(Q2)는 RDF description 데이터를 검색하는 data query 중에서 주어진 클래스 URI의 인스턴스를 검색하는 질의이다. 세 번째(Q3)와 네 번째 질의(Q4)는 SFW query이다. 세 번째 질의는 RDF description 데이터에 대한 경로를 포함하는 질의이고, 마지막 질의는 RDF schema 데이터 경로를 포함하는 질의이다.

5.1 질의 처리 성능

본 실험에서는 표 5.1의 질의를 RDF description 데이터의 사이즈가 500K, 1M, 2M인 데이터베이스에 대해 수행하였다. 각각의 데이터의 RDF schema는 동일하다. 표 6은 실험 결과를 보여준다. 실험 결과에서 Q1과 Q4에 대한 수행 시간은 데이터의 사이즈와 상관없이 일정함을 볼 수 있다. 이는 Q1과 Q4가 RDF schema 데이터에 대한 질의이고, 실험에 쓰인 데이터의 schema 데이터가 일정하기 때문이다. Q2와 Q3의 경우 RDF description 데이터의 사이즈가 클수록, 질의 처리 시간이 길어짐을 볼 수 있다. 특히, Q3의 경우 데이터의 사이즈가 커짐에 따라 처리 시간이 4배정도씩 길어짐을 볼 수 있는데, 이는 RDF description 데이터에 대한 SFW 질의를 XPath 질의로 변환할 때, 여러 XPath 질의들로 변환되고, 이에 따라 조인이 많이 필요하기 때문에 그만큼 시간이 길어진다.

표 4. RQL 질의에 대한 처리 시간 (단위: 초)

	500K	1M	2M
Q1	0.137	0.134	0.133
Q2	1.802	2.874	4.447
Q3	44.384	154.863	610.918
Q4	0.438	0.441	0.437

5.2 RDFSuite와의 성능 비교

그림 3은 동일한 데이터에 대한 질의 처리 시간을 RDFSuite 시스템과 비교한 결과이다. Q2의 경우 RDFSuite 시스템과 질의 처리 시간이 거의 비슷하지만, 다른 질의의 경우 본 연구에서 구현한 시스템이 좀 더 나은 성능을 보임을 볼 수 있다. 특히, Q3의 SFW 질의의 경우, 앞 절에서는 데이터 크기가 커짐에 따라 수행 성능이 나빠짐을 볼 수 있었지만, 다른 시스템과 비교했을 경우, 나쁘지 않은 성능을 보여 주고 있다. 이는 본 연구에서 보다 효율적으로 조인 연산을 수행하고 있기 때문이다.

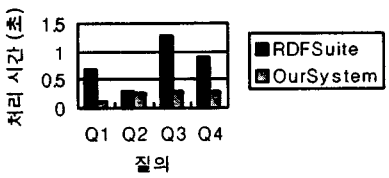

그림 3. RDFSuite 시스템과의 질의 처리 성능 비교

6 결론

본 연구에서는 RDF description 데이터와 RDF schema 데이터를 XML 데이터베이스 시스템에 저장하고 이를 검색하는 방법을 제안하였다. 시멘틱 웹이 차세대 웹으로 자리잡기 위해서는 가장 먼저 RDF 기반 기술들이 정립되어야 하고, 기반 기술들 중에서 가장 중요하고 시급한 문제는 RDF 데이터를 저장하고 검색하는 기술이다. 본 연구에서는 RDF 데이터를 XML 데이터베이스 시스템에 저장함으로써, 시멘틱 웹의 주된 데이터인 XML 데이터와 이에 대한 메타 데이터인 RDF 데이터를 통합적이고 효율적으로 다룰 수 있는 방법을 제공한다. 또한, RDF 질의 언어인 RQL 질의 언어를 통해 RDF description 데이터 뿐만 아니라, RDF schema 데이터도 검색할 수 있도록 함으로써, RDF 데이터 모델에 맞게 원하는 데이터를 쉽게 검색할 수 있는 방법을 제공한다. 또한, 저장할 때 RDF schema의 클래스와 속성의 계층 구조를 저장함으로써, 보다 쉽고 빠르게 이 데이터에 대한 RQL 질의를 처리하도록 하였다. 검색할 때는 최대한 데이터베이스 시스템에서 질의를 처리하도록 하였고, 또한 보다 효율적인 조인 기법을 제안하고 있다. 본 연구에서 제안한 여러 기법들의 성능 평가에서도 RDFSuite 시스템보다 질의 처리 성능이 좋음을 확인할 수 있었다.

본 연구에서 제안하고 있는 저장 및 검색 기법들은 지속적인 연구를 통해 보다 효율적인 방법으로 발전할 수 있을 것이다. 하지만, 아직 정립되지 않은 RDF 데이터 저장 및 검색 방법을 XML 데이터베이스 시스템을 사용하여 제안함으로써 앞으로의 RDF 데이터를 다루는 여러 연구들의 새로운 방향을 제시한 데에 본 연구의 의의가 있다고 할 수 있다.

참고 문헌

[1] O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation, 1999

[2] S. Alexaki, V. Christophides, G. Karvounarakis, D. Plexousakis, K. Tolle. The RDFSuite: Managing Voluminous RDF Description Bases, Technical Report, ICS-FORTH

[3] G. Karvounarakis, S. Alexaki, V. Christophides, D. Plexousakis, M. Scholl. RQL:A Declarative Query Language for RDF. WWW2002

[4] D. Brickley and R.V. Guha. RDF Vocabulary Description Language 1.0: RDF Schema. W3C Working Draft, 2002

[5] J. Clark, S. DeRose. XML Path Language (XPath) Version 1.0. W3C Recommendation 1999

[6] D. Beckett. RDF/XML Syntax Specification (Revised). W3C Working Draft 2002